

Il Calcolatore, motore dell'azione didattica nell'insegnamento-apprendimento della matematica

Francesco Costabile

Dipartimento di Matematica LAN-CIRD - Università della Calabria
costabile@unical.it

Sommario

Il Calcolatore è sorgente inesauribile e risolutore efficace di problemi; in un quadro metodologico opportuno, può diventare motore essenziale e continuo dell'azione didattica nell'insegnamento-apprendimento della Matematica. Infatti il Calcolatore attraverso la programmazione obbliga ad un "training mentale", ma anche consente l'effettiva risoluzione o simulazione di problemi della realtà sensibile. Entrambi questi fattori sono obiettivi fondamentali nell'insegnamento-apprendimento della Matematica. In questo percorso le difficoltà maggiori si incontrano nella comunicazione uomo-macchina, ovvero nel linguaggio di programmazione; a tal fine si propone un nuovo linguaggio denominato MatCos. Esso, in lingua italiana, presenta una struttura sintattica semplice; è orientato alla Matematica ed è modulare, ogni modulo riferentesi ad una precisa fascia d'età. Tutto ciò sarà illustrato con opportuni esempi.

1 Il quadro metodologico di riferimento

“Dobbiamo confessare che la matematica ha costituito la tortura degli scolari di tutto il mondo e che l'umanità ha tollerato questa tortura per i suoi figli come una sofferenza inevitabile per poter conquistare delle cognizioni necessarie. Ma l'insegnamento non deve mai essere una tortura, e non saremo dei veri professori se non faremo in modo, con tutti i mezzi, di trasformare questa sofferenza in godimento; godimento che non vuole dire mancanza di ogni sforzo ma che, anzi, deve essere il frutto di un lavoro svolto con passione e gioia.” (Pedro Puig. ADAM)

Convegno internazionale sull'insegnamento della matematica, Madrid 1957

“Al giorno d'oggi, la matematica bella e fatta (la matematica morta) domina nelle scuole e nell'Università. Il calcolatore ci rammenta con forza che la matematica è un'attività. Si assiste a una ripresa della risoluzione di problemi. Ogni programma è la soluzione di un problema, ma a un livello superiore a quello della soluzione di un enigma o di un indovinello. Un programma crea un'azione. Esso innesca un processo reale e lo dirige” (Artur

Engel)

III seminario C.I.M.E Echterrich 1973

“Se si insegna ad un bambino a programmare in qualche linguaggio informatico, questo esercizio logico lo renderà padrone e non schiavo del computer” (Umberto Eco) “La Repubblica”, 8/1/2000

Queste tre fasi delineano assai bene la “filosofia” di questa conferenza. Adam, nel seguito della sua relazione, pose l’accento sulla necessità di non limitare l’insegnamento della Matematica alla sola fase operatoria di relazioni fra concetti puri ed enti idealizzati, perché la stragrande maggioranza degli allievi che svilupperà attività matematica, affronterà problemi del mondo concreto in cui gli enti non sono già depurati e quindi simili a quelli con i quali si era già abituato; occorrerà, al contrario, che l’apparato logico-astratto, che è una fase intermedia ed importante, nella risoluzione dei problemi quantitativi della filosofia naturale, sia preceduto da una fase di schematizzazione o di astrazione in cui la mente riduce a schemi matematici i fenomeni naturali oggetto di studio, e che sia seguito da un’altra fase che diremo di <<concrezione>> ovvero di interpretazione, di proiezione dei risultati ottenuti, nel campo della realtà. Il procedimento proposto da Adam oggi giorno è più conosciuto come Problem-solving o modellizzazione della realtà sensibile e può essere schematizzato come in fig. 1.

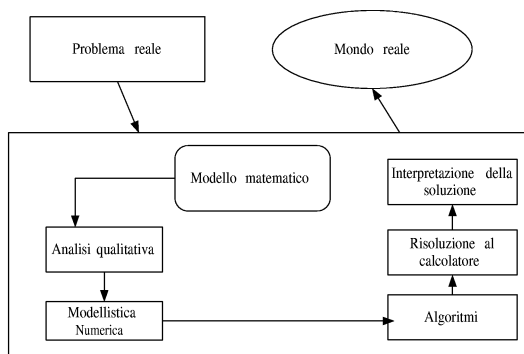


Fig. 1

Naturalmente nulla esclude che nell’analisi qualitativa e numerica si possa giungere ad ulteriori generalizzazioni ed astrazioni, che conducano ad una fase più specificatamente operatoria di relazioni tra enti idealizzati; ciò è anche parte integrante ed essenziale del pensiero matematico perché contribuisce in maniera fondamentale alla formazione logico mentale.

Se ci si fermasse, però, alla fase puramente operatoria di relazioni tra enti astratti si correrebbe il rischio, molto forte, di presentare la matematica bella e fatta, con poche probabilità di successo, ovvero che l’essenza venga accettata e recepita.

Possiamo, dunque, riassumere il quadro metodologico proposto affermando che l’attività matematica scolastica deve essere principalmente fi-

nalizzata al “trainig mentale” e al “Problem-solving”, inteso principalmente come connessione matematica-realtà sensibile. In questo quadro si inserisce assai bene l’uso del calcolatore attraverso la programmazione.

2 Il calcolatore

Il calcolatore elettronico, digitale, automatico, per uso generale, cui intendiamo riferirci è senza dubbio alcuno l’innovazione tecnologica del secolo scorso che ha maggiormente influenzato lo sviluppo socio culturale. Questa macchina, a differenza di tutte le altre esistenti, ha capacità di dialogo con l’uomo, che ne dirige in ogni caso il processo operativo.

E’ proprio questa caratteristica quella che ne ha determinato il successo e quindi l’applicazione nei più svariati contesti della realtà sensibile. Un dialogo, però, non facile dal momento che i due soggetti, uomo e calcolatore, hanno linguaggi, l’uno detto naturale, l’altro di base, notevolmente diversi. Si pose, perciò, sin dall’inizio il problema della creazione di linguaggi intermedi tra i due e che fossero finalizzati a particolari applicazioni. Sorsero così il Fortran per i calcoli scientifici, e poi il Cobol per le applicazioni commerciali; il Pascal più flessibile del Fortran e così via fino ad arrivare ai più moderni C++ e Java che interagisce con la rete internet. I precedenti sono tutti linguaggi di programmazione generali se pur, più o meno, convenienti per particolari applicazioni. Non potevano, però, mancare linguaggi particolarmente dedicati, ad esempio, alla matematica: Mathematica, Maple, Derive. Questi software noti più propriamente come computer algebra system (CAS) sono in grado di effettuare manipolazioni simboliche algebriche e di fornire con poche istruzioni la soluzione analitica, grafica e numerica di problemi matematici. Più specificatamente rivolti a rappresentazioni geometriche sono i software Cartesio, Get, Cabri-geomètre, Stretch-Pad. L’uso di questi software nella pratica didattica pone, però, grossi interrogativi di natura pedagogica, ovverosia di una loro effettiva utilità nel delicato processo di insegnamento-apprendimento della matematica in quanto forniscono, nella maggior parte dei casi, la soluzione completa o quasi del problema. Naturalmente molto dipende dal contesto, dalla metodologia e dagli obiettivi formativi preposti. Certamente diversa è, infatti, la situazione dello la situazione dello studente universitario di matematica e non, dallo studente della scuola secondaria. Comune, però, ad entrambe le situazioni è di sicuro una nuova metodologia di insegnamento, se si vuole fare uso dei software esistenti.

La critica principale che si fa all’uso dei CAS è che essi presentano nella maggior parte dei casi la soluzione bella e fatta, senza richiedere alcuno sforzo mentale o attività intellettuale che dir si voglia, e contemporaneamente richiedono l’uso di un linguaggio con una sintassi pesante e certamente poco adeguata alla scuola secondaria.

Questa osservazione è largamente condivisibile specialmente se si accetta il quadro metodologico e gli obiettivi formativi dinanzi esposti. Infatti

abbiamo parlato di “training mentale” nell’attività matematica al fine di offrire allo studente la possibilità di sviluppare al meglio il suo talento intellettuale.

Per ottenere questo occorre non rinunciare alla programmazione, perché programmare è un modo di pensare, pensare con un fine, ovvero risolvere un problema. D’altra parte l’importanza della programmazione nel processo di insegnamento/apprendimento è stata evidenziata varie volte sia in contesti nazionali che internazionali ([1], [18]), per cui in questa sede ci limitiamo a riaffermarne il principio e ad indicare un percorso in cui essa diventa parte essenziale dell’azione didattica. La difficoltà principale che si incontra nell’usare il calcolatore in modo consapevole e sistematico ovvero programmando in un corso di matematica a qualsiasi livello scolastico, è costituita dal linguaggio di programmazione, che spesso può essere poco adatto a precisi contenuti e precise fasce d’età e quindi contesti educativi. Su questa base nasce il progetto MatCos, più precisamente esso ha il duplice scopo:

- avviare lo studente alla programmazione utilizzando i concetti matematici adeguati alla sua età;
- far apprendere ed interiorizzare metodi e concetti matematici sfruttando le potenzialità del computer.

3 L’ambiente MatCos

Ambiente integrato che consente lo sviluppo e l’esecuzione di programmi rivolti principalmente alla matematica, fruibile su PC in ambiente Windows. Utilizza un linguaggio di programmazione ad alto livello denominato anche MatCos, linguaggio espresso in lingua italiana che conserva legami con quello naturale, ed è molto vicino al linguaggio matematico. MatCos è diviso in moduli, ciascuno dei quali è rivolto ad una precisa fascia d’età scolare, il linguaggio di programmazione è composto da blocchi di istruzione per i quali si rinvia, per una esauriente discussione, ai manuali in bibliografia.

Si menziona solo la caratteristica fondamentale dell’ambiente MatCos che consiste nel poter eseguire ogni programma “ passo-passo”, ovvero un comando alla volta, visualizzando così i risultati intermedi.

Questa peculiarità è molto importante dal punto di vista didattico perché dà allo studente la possibilità di controllare ogni passo dell’algoritmo e correggere più facilmente eventuali errori.

4 Alcuni esempi

Si riportano a chiarimento di quanto sopra esposto, alcuni esempi esplicativi con precisi obiettivi, secondo l’utente cui si rivolge.

Esempio 1 - Fascia d’età 8-11 anni - Percorso geometrico: dal punto al poligono e al cerchio.

I comandi necessari sono:

```
punto; linea; linea(punto,punto);
segmento(punto,punto); segmento(punto,numero).
```

Possono essere utilizzati a seconda del “gusto” didattico sia in fase preventiva-esplorativa che in fase successiva di approfondimento e verifica. Ad esempio i comandi `punto`; `linea`; consentono rispettivamente di tracciare col mouse un punto e una linea, quindi stimolare l'intuizione di infiniti punti e punti allineati.

Il comando `linea(punto, punto)` consente, invece, di tracciare una linea tra due punti assegnati e pone il problema di quante linee si possono condurre e quale sia quella che fornisce il “cammino” più breve tra i due punti. Ne scaturisce il concetto di segmento e quindi il comando `segmento(punto,punto)`;

Tutto ciò si realizza col semplice segmento di programma **MC1** il cui output è in fig. 2.

```
MC1    Punto;
        Linea;
        A=Punto;
        B=Punto;
        Linea(A,B);
        s=Segmento(A,B);
```

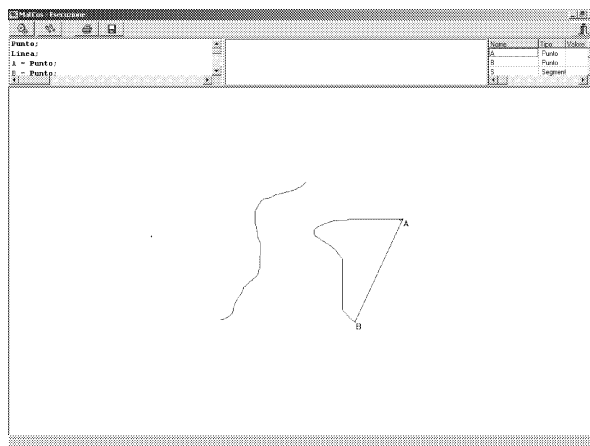


Fig. 2

Dal concetto di segmento si può passare a quello di triangolo e al semplice programmino che lo realizza:

MC2 A=Punto;
 B=Punto;
 C=Punto;
 Segmento(A,B);
 Segmento(B,C);
 Segmento(A,C);

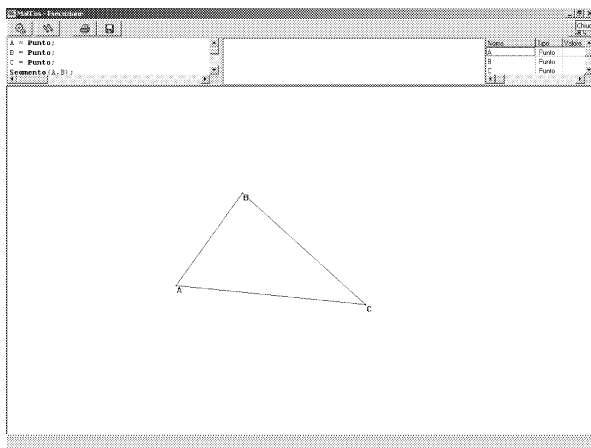


Fig. 3

Dopo aver constatato la lunghezza di un segmento; il comando **segmento (punto,numero)**; consente di tracciare un segmento avendone fissato un estremo e la lunghezza. Si prende atto, in questo modo delle diverse direzioni del piano e della necessità di fissare due punti per determinarne univocamente una sola. Non solo ma lo stesso comando conduce in maniera naturale alla rotazione, ai due versi di rotazione possibili e alla circonferenza (Fig. 4).

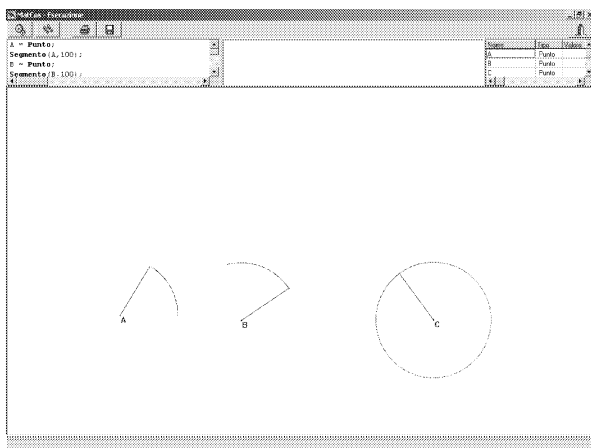


Fig. 4

Esempio 2 - Fascia d'età 11-13 anni - Un percorso aritmetico-statistico.

L'algoritmo euclideo della divisione tra naturali porta al concetto di multipli e divisori e quindi in modo spontaneo al problema del calcolo dei divisori di un numero assegnato. Le istruzioni MatCos che forniscono il quoziente intero e il resto sono rispettivamente

DIV e RDIV

Allora il semplice programma seguente calcola tutti i divisori di un numero assegnato:

```
MC3      N = LeggiNum("numero di cui vogliamo i divisori");
          PER (i DA 1 A n) ESEGUI;
            r=n RDIV i;
            SE ( r = 0 ) ALLORA
              Stampa( i, "e' divisore" ) ;
            ALTRIMENTI
              Stampa ( i, "non e' divisore");
          FINE;
```

Dal concetto di divisore si passa facilmente a quello di numero primo e al problema naturale: dato un numero stabilire se esso è primo. Tale problema, anche se non in modo ottimale, può essere risolto col seguente programma che è una leggera variazione del precedente.

```
MC4      n=LeggiNum("numero dato");
          PER ( i DA 2 A ( n - 1 )) ESEGUI;
            r=n RDIV i ;
            SE ( r=0 ) ALLORA ESEGUI;
              Stampa(n, "non e' primo");
              Stop;
            FINE;
          FINE;
          Stampa(n, "e' primo");
```

Con i primi elementi di statistica si perviene sempre alla rappresentazione di dati con opportuni diagrammi; MatCos dispone di semplici istruzioni per la rappresentazione:

```
DiagCirc (<lista numeri>);
Istogramma (<lista numeri>);
DiagStr (<lista numeri>);
```

che sono illustrati in fig. 5.

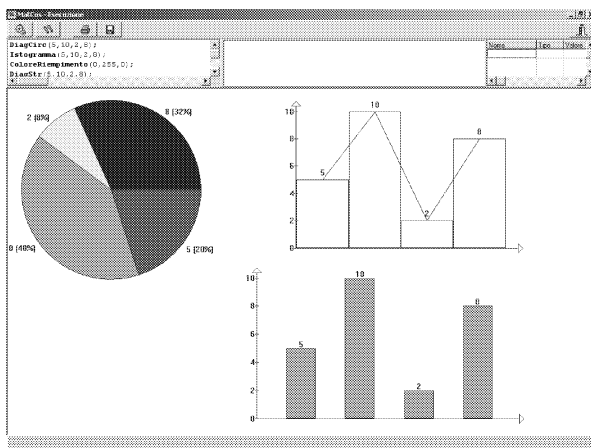


Fig. 5

Esempio 3 - Fascia d'età 11-15 anni - Percorso geometrico: Costruzione di poligoni regolari.....isometria

Il seguente programma costruisce un quadrato, lo trasla secondo un vettore assegnato, quindi lo ruota di 60 gradi intorno ad un suo vertice

```
MC5  A=Punto;
      B=Punto;
      r=Retta(A,B);
      p=Perpendicolare(r,A);
      Cancella(r);
      m=Distanza(A,B);
      s=Segmento(A,m,p);
      Cancella(p);
      D=s.Estremo(2);
      C1=Circ(B,m);
      C2=Circ(D,m);
      C=Intersezione(C1,C2);
      q=Poligono(A,B,C,D);
      t1=Segmento_Or(Punto,Punto);
      Cancella(C1,C2);
      q1=Trasla(q,t1);
      ColorePenna(0,128,0);
      q2=Ruota(q1,A,60,ORARIO);
```

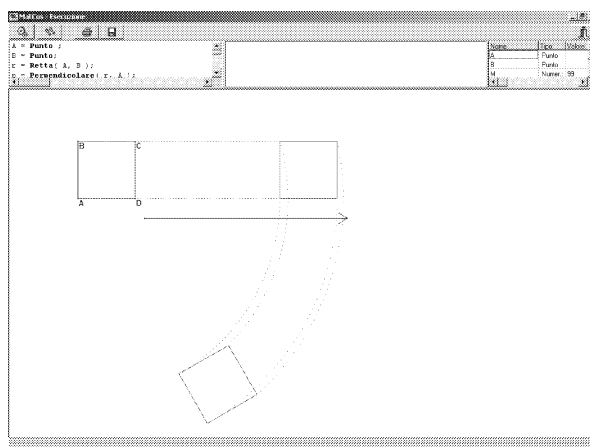



Fig. 6

Esempio 4 - Fascia d'età 15-18 anni. Percorso analitico.

A) *Alla scoperta dell'equazione della retta e delle condizioni di parallelismo e di perpendicolarità* Il seguente programma introduce il concetto di equazione della retta in quanto porta alla intuizione della relazione che sussiste tra le coordinate dei punti di una stessa retta.

```
MC6      RifCart;
        A=Punto(2,5);
        B=Punto(1,3);
        t=Lista;
        m=Matrice(10,2);
        r=Retta (A,B);
        PER ( i DA 1 A 10 ) ESEGUI;
          Colore(i);
          T(i) = PuntoACaso_su(r);
          m(i,1) = T(i).x;
          m(i,2) = T(i).y;
        FINE;
        StampaMatr(m);
```

Dopo aver eseguito varie volte questo programma si può constatare che tra le coordinate dei punti di una retta non parallela all'asse y , c'è una relazione del tipo $y = mx + n$. La dimostrazione rigorosa, necessaria, segue, poi, con tecniche standard.


```

MC8([14])    rifcart;
              Stampa("Il programma permette di studiare il
                  comportamento di una funzione in un intorno
                  simmetrico di un punto");
              f=leggifunz("introdurre la funzione");
              x0=legginum("punto limite");
              C=punto(x0,0);
              colore(8);
              Stampa ("Per evidenziare lo studio della funzione
                  di un intorno del punto si traccera' il grafico
                  della retta di equazione x=", x0);
              s=rettapar_asse("y", x0);
              stampa ("Il grafico della retta verra' cancellato
                  dopo aver calcolato le coordinate dei punti
                  della curva richiesti");
              n=10;
              ms=matrice (n, 2);
              md=matrice(n, 2);
              h=1/n ;
              xs=x0-n*h;
              xd=x0+n*h;
              PER (i DA 1 A n) ESEGUI;
                  ys=valutafunz(f, xs);
                  ms(i, 1)= xs;
                  ms(i, 2)= ys;
                  colore (4);
                  punto(xs, ys);
                  xs=xs+h;
                  yd=valutafunz(f, xd );
                  md(i, 1)=xd;
                  md(i, 2)=yd;
                  colore(3);
                  punto(xd, yd);
                  xd=xd-h;
              FINE;
              cancella(s);
              stampamatr(ms);
              stampamatr(md);
              stampa("grafico della funzione nell'intorno del
                  punto limite");
              colore(7);
              graficofunz(f, x0-1, x0+1);
              stampa("grafico della funzione");
              graficofunz(f);

```

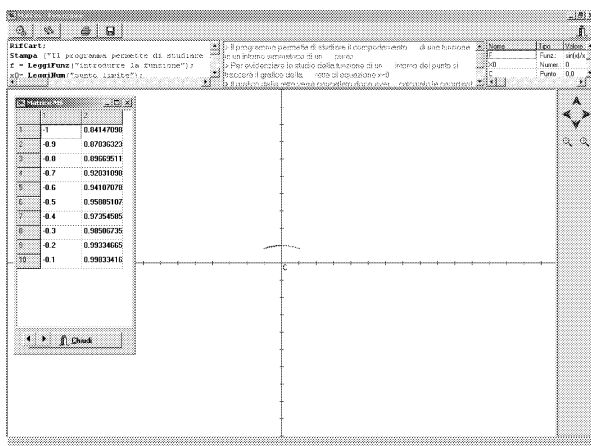


Fig. 8

Nella fig. 8 è riportato l'output relativo alla funzione $y = \frac{\sin x}{x}$ e $x_0 = 0$.

C) *Studio delle proprietà qualitative, quantitative e rappresentazione grafica di una funzione reale di variabile reale.* Tradizionalmente nella scuola secondaria tale argomento occupa quasi un intero anno di lavoro. Ora tutti i CAS, ma anche MatCos, sono in grado di fornire la soluzione in pochi secondi: che fare dunque? Ignorare questi strumenti tecnologici e continuare, come se non ci fosse niente di nuovo; oppure accettare passivamente i CAS e concentrare il lavoro di un anno in pochi minuti?

Naturalmente entrambe le soluzioni rappresentano casi estremi e come tali vanno mediati. Una azione didattica mediatrice può avere la seguente linea: dopo il concetto di funzione reale di variabile reale si introduce la rappresentazione grafica per punti attraverso un CAS. Ad esempio in MatCos sono sufficienti le seguenti istruzioni:

```
MC9    Rifcart;
        f = Funzione("(1+x)/x^2");
        GraficoFunz( f );
```

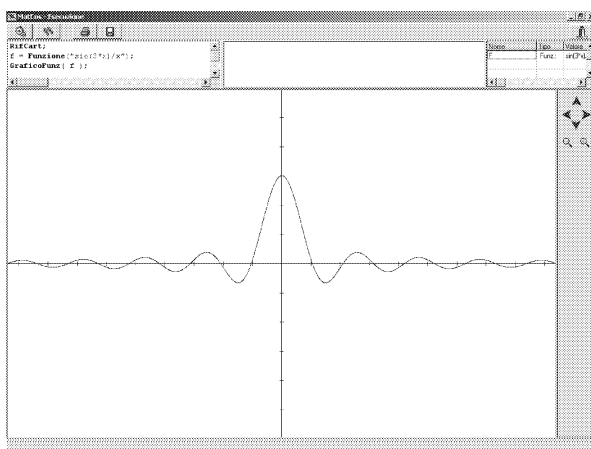


Fig. 9

Dalla rappresentazione grafica è facile passare alla descrizione intuitiva delle proprietà qualitative (monotonia, punti estremali, zeri, comportamento all'infinito o nell'intorno di particolari punti). Per ottenere valori quantitativi basta considerare l'istruzione

```
ValutaFunz(f, x);
```

che dà il valore della funzione f nel punto assegnato x ; mentre il comando

```
fzero (f, a1, a2, eps);
```

fornisce lo zero della funzione f nell'intervallo $(a1, a2)$ con la tolleranza eps . Dopo questa fase esplorativa si può passare alla formalizzazione simbolica e alla determinazione analitica delle condizioni che giustificano le proprietà intuitive della fase precedente.

In questo modo lo studente non solo non perde consapevolezza, utilizzando gli strumenti tecnologici avanzati ma apprezza e acquista gusto per l'attività matematica (fase simbolica operatoria) quale strumento essenziale per la ricerca (o la giustificazione) di alcune "verità". MatCos mette a disposizione anche il comando

```
DerivataFunz (f);
```

per il calcolo della derivata analitica della funzione, quindi per la stampa il comando

```
StampaFunz (f);
```

E' facile, quindi, mettere in relazione il grafico di una funzione e della sua derivata, col semplice programmino

```

MC10  RifCart;
      f = Funzione("cos(2*x)");
      GraficoFunz( f );
      ColorePenna(0, 255, 0);
      g=DerivataFunz( f );
      GraficoFunz( g );
      StampaFunz( g );

```

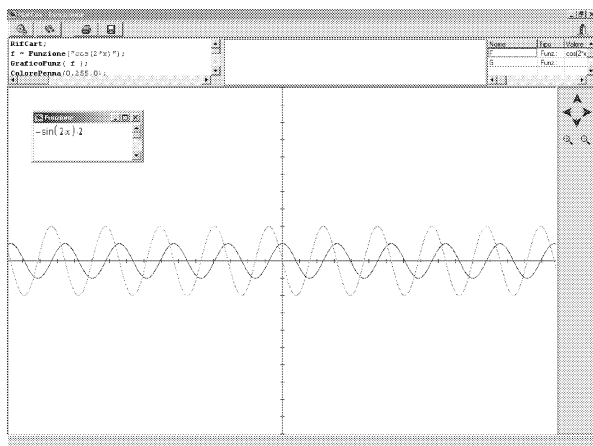


Fig. 10

Esempio 5-Fascia d'età ≥ 19 anni - Soluzione immediata di alcuni problemi.

MatCos è anche in grado di fornire la soluzione di alcuni problemi che si incontrano più frequentemente nelle applicazioni. Ad esempio:

```
M = Inversa(N, eps);
```

fornisce la matrice inversa della matrice N, a meno di eps;

```
d = Determinante(N);
```

fornisce il determinante della matrice N;

```
x = SistemaLineare(A, b, eps);
```

fornisce la soluzione del sistema lineare di matrice A, termine noto b, a meno di eps;

```
A = Autovalori(N, eps);
```

fornisce gli autovalori della matrice N a meno di eps;

```
f = Primitiva(g);
```

fornisce una primitiva della funzione g ;

```
n = Integrale (f, a, b, eps);
```

fornisce l'integrale definito della funzione f tra a e b a meno di ϵ .

```
f = PolT(g, a1, a2);
```

fornisce il polinomio di Taylor relativo alla funzione g , di punto iniziale a_1 e grado a_2 ;

```
n = Fattoriale(m);
```

fornisce il fattoriale di m ;

```
n = Binomiale(a1, a2);
```

fornisce il coefficiente binomiale $\binom{a_1}{a_2}$. Si può, anche, ottenere il grafico di una funzione a due variabili, ad esempio:

```
MC11  Rifcart3D;  
       ColoreRiempimento(255,0,255);  
       F=Funzione3D("sin(x*y)");  
       Graficofunz3D( F );
```

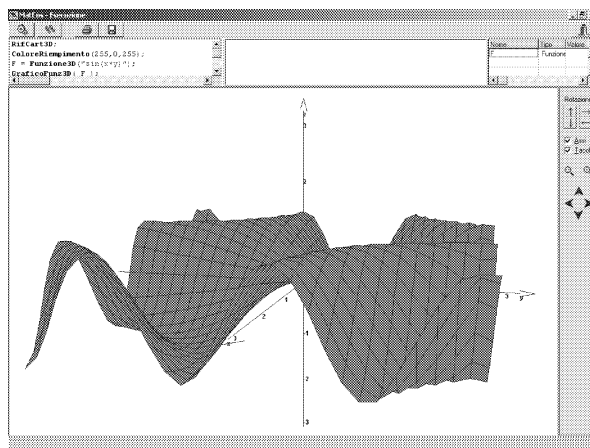


Fig. 11

Fornisce il grafico della funzione $\sin(x * y)$.

Esempio 6 - Matematica e realtà sensibile.

Nel quadro metodologico esposto all'inizio il legame Matematica-realtà sensibile è ritenuto importante, MatCos è di notevole aiuto per esplicitare questo rapporto anche nella scuola secondaria. Ad esempio osservando un comune ventaglio si può descrivere un modello matematico con relativo algoritmo, la cui implementazione in MatCos fornisce l'output di Fig.12

```
MC12([13])  Colore(4);
            s=SEGMENTO(PUNTO, PUNTO);
            C=PUNTO_SU(s);
            A=PUNTO_SU(s);
            B=PUNTO_SU(s);
            AB=SEGMENTO(A, B);
            AC=SEGMENTO(A, C);
            ALFA=LEGGINUM;
            I=0;
            ESEGUI FINQUANDO(I<=ALFA);
                RUOTA(s, C, I, ORARIO);
                I=I+1.5;
            FINE;
            I=0;
            Colore(9);
            ESEGUI FINQUANDO(I<=ALFA);
                RUOTA(AB, C, I, ORARIO);
                I=I+1;
            FINE;
```

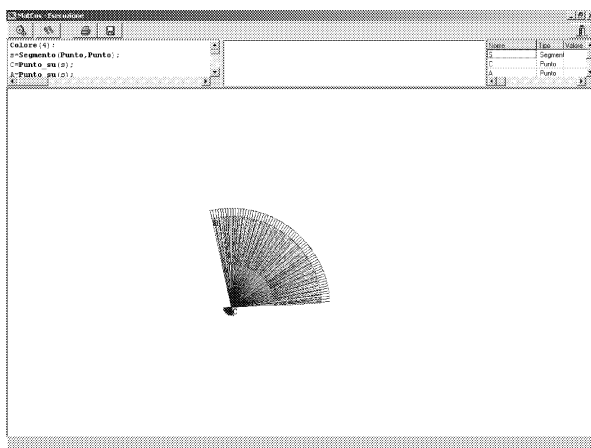


Fig. 12

5 Quanto è affidabile MatCos?

E' difficile stabilirlo, ma in fig.13 e fig.14 è riportato il grafico della derivata della funzione

$$y = e^{|x-1|} + 2x \text{ rispettivamente eseguito con MatCos:}$$

MC13 RifCart;
 f=Funzione("Exp(Abs(x-1))+2*x");
 g=DerivataFunz(f);
 GraficoFunz(g);

e Mathematica:

```
f[x_] = Exp[Abs[x-1]]+2*x
g[x_] = D[f[x], x]
Plot[g[x], x, -4, 4]
```

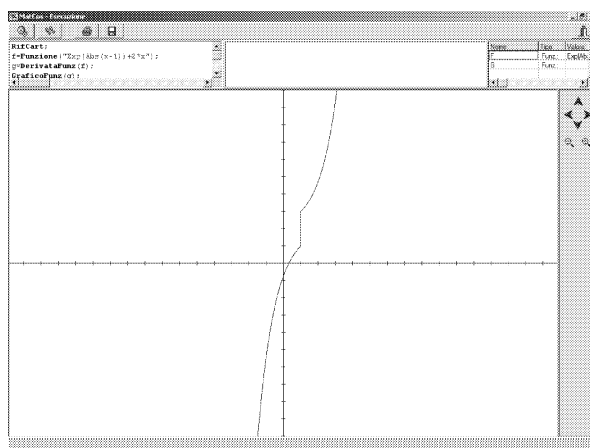


Fig. 13 - Grafico con Matcos

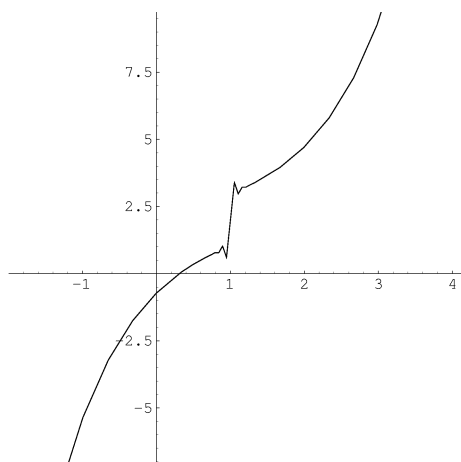


Fig. 14 - Grafico con Mathematica

Il commento è lasciato al lettore.

6 Sperimentazione e conclusioni

Da alcuni anni il Progetto MatCos è in fase di sperimentazione in un campione di istituti secondari sia di primo grado che di secondo grado della Regione Calabria.

Più in particolare sono stati sperimentati i moduli 1.1, 1.2, 1.3 relativi alla scuola media, e i moduli 1.4, 1.5, 1.6 per la scuola superiore; in questo anno scolastico si sta sperimentando il modulo 1.7 per la IV classe delle superiori. La sperimentazione si svolge sotto la direzione del Centro Interdipartimentale di Ricerca Didattica (CIRD) dell'Università della Calabria e prevede tre stage di formazione degli insegnanti sperimentatori e due test di verifica per gli allievi. I risultati dei test sono stati monitorati in ([11],[12]) e vengono usualmente presentati annualmente nel corso di un seminario pubblico. La reazione dei docenti sperimentatori e soprattutto degli allievi è stata ampiamente positiva, sia sul piano dell'interesse suscitato che su quello dei contenuti. Si è potuto notare un apprendimento più partecipato, consapevole e motivato. Molti studenti hanno presentato lavori autonomi, naturalmente riferiti all'età, che erano dei veri progetti applicativi, in cui concetti matematici venivano elaborati per simulare, attraverso il computer, oggetti della realtà sensibile; alcuni di questi si possono consultare sul sito: **cird.unical.it**.

Va anche detto che il linguaggio MatCos o meglio il suo modulo 1.x, è stato sperimentato dal sottoscritto nel primo anno di studi universitari sia in corsi di tipo Istituzioni di Matematiche per Scienze Biologiche, Farmacia etc. che per studenti di Matematica nel modulo di Laboratorio di Programmazione e Calcolo. L'impatto è stato positivo sia sul piano problematico, culturale e professionale che su quello di primo approccio scientifico al calcolatore. Concludendo possiamo affermare senza tema di smentita, che l'uso

costante e continuo del calcolatore nell'azione didattica dell'insegnamento-apprendimento della Matematica non solo è possibile ma è utile e proficuo, in entrambi i versi, naturalmente modificando il quadro metodologico di insegnamento-apprendimento tradizionale.

Riferimenti bibliografici

- [1] Biehler R., Scholz R., Strasser R., *Didactis of Mathematics as a Scientific Discipline*, Kluver, London, 1994.
- [2] Costabile F., *Computer e insegnamento-apprendimento della matematica*, Educazione matematica e sviluppo sociale, a cura di N. Malara, Rubbettino ed., Soveria M. (CZ), 2002.
- [3] Costabile F., *MatCos 1.1, 1.2, 1.3, 1.4, 1.5, 1.6, 1.7, 1.X*, Calìo Informatica, Rende (CS), 1999–2002.
- [4] Costabile F., *Come va il Piano Nazionale d'Informatica*, Periodico di Matematiche, 1997.
- [5] Costabile F., *Matematica e Computer: un binomio ormai scisso*, Periodico di Matematiche.
- [6] Costabile F., *Algebra per il biennio 1 e 2; Geometria per il biennio 1 e 2*, Liguori ed., Napoli, 1988.
- [7] Costabile F., *Matematica tre, quattro, cinque*, Liguori ed., Napoli, 1990.
- [8] Costabile F., *Lezioni di Istituzioni di Matematiche mod. 1, 2, 3*, Liguori ed., Napoli, 1998.
- [9] Costabile F., Serpe A., *Matematica e Computer con MatCos*, Progetto Alice, 2002, pp. 79-98.
- [10] Costabile F., Serpe A., *MatCos: un nuovo strumento per fare Matematica*, ADT, 4 Convegno Nazionale, Monopoli 11-13/10/2002.
- [11] Costabile F., Serpe A., *Matematica e Computer: l'esperimento nella scuola media*, CIRD-UNICAL, Pubbl. n. 3, 2002.
- [12] Costabile F., Serpe A., *Matematica e Computer : L'esperimento nel biennio della Scuola Superiore*, CIRD-UNICAL), Rapporto n. 34, 2002.
- [13] Costabile F., Golemme R., *Matematica e realtà sensibile*, CIRD-UNICAL, Pubbl. n. 2, 2001.
- [14] Costabile F., Gaccetta C., *Comportamento di una funzione in un punto*, CIRD-UNICAL, Pubbl. n. 4, 2002.

- [15] Kutzeler B., *Quale matematica dobbiamo insegnare, quando insegniamo con il CAS?*, ADT, 4 Convegno Nazionale, Monopoli 11-13/10/2002.
- [16] Kutzeler B., *The Algebraic Calculator of Pedagogical tool for teaching Mathematics*, The International Journal for computer Algebra in Mathematics Educational, 1, 2000, pp. 5-24.
- [17] Dodaro N., Baroncini P., Trezzi D., *Elementi di Matematica I, II, III, IV, V*, Ghisetti e Corvi, Milano, 1991.
- [18] Reggiani M., *Insegnare a programmare nella scuola media inferiore obiettivi, risultati, difficoltà riflessioni.*, L'insegnamento della Matematica e delle Scienze integrate 17B, 1, 1994, pp. 65-91.
- [19] Perotti A., Lorenzi A., Baccelli B., *Analisi Matematica con Derive*, McGraw Hill, 1992.
- [20] Boieri P., Gremillari D., Keiton M., *Cabri - geometre*, Texas Instrument, 1994.